UNCLASSIFIED MASTER COPY - FOR REPRODUCTION PURPOSES 2

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

DTIC ELECTE FEB 1 5 1988 H D

AD-A204 752

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
...HEDULE	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
...UMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) ARO 22935.2-EL-S

6a. NAME OF PERFORMING ORGANIZATION Technical Solutions, Inc.	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION U. S. Army Research Office
6c. ADDRESS (City, State, and ZIP Code) P.O. Box 1148 Mesilla Park, NM 88047		7b. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U. S. Army Research Office	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAG29-85-C-0026

8c. ADDRESS (City, State, and ZIP Code)
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Automatic Documentation Methodologies for Software Maintenance

12. PERSONAL AUTHOR(S)
L. D. Landis, P. M. Hyland, A. L. Gilbert, A. J. Fine

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 8/1/85 TO 11/15/88	25 Jan 89	4

16. SUPPLEMENTARY NOTATION The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Software Maintenance, Maintenance Programmers, Programmers, Program Languages. JES

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Work on this project led to some interesting conclusions. The first of these was the determination that none of the methodologies in use today is really optimal in its utility to maintenance programmers. While NS Diagrams and others are familiar to programmers and to many knowledgeable users, the optimal tool would provide a new methodology that provided graphics, text, detail, overview, static structure, and dynamic dataflow all in a visually-appealing, user-friendly way.

(continued on back)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

The second conclusion is that the generality requirement is indeed a requirement, not just a consideration of interest. Programmers working with different languages have widely different perceptions about what is useful documentation. Thus a documentation tool that provided only one type of documentation would not solve the problem.

ARO 22935.2-EL-S

AUTOMATIC DOCUMENTATION METHODOLOGIES FOR SOFTWARE MAINTENANCE

FINAL REPORT

L. D. LANDIS
P. M. HYLAND
A. L. GILBERT
A. J. FINE

15 JANUARY 1989

U.S. ARMY RESEARCH OFFICE

CONTRACT #DAAG029-85-C-0026

Technical Solutions, Inc.
P.O. Box 1148
Mesilla Park, N.M. 88047

STATEMENT OF THE PROBLEM STUDIED

Overview Of The Problem:

Software has a limited lifetime of usefulness, because as existing software ages, support becomes more difficult. Major factors in determining when to replace rather than maintain software are 1) the cost of training new personnel to maintain that software, and 2) the time required to train those personnel. Modification or repair is made easier by accurate documentation. By providing automatic techniques of documentation generation, the useful lifetime of a software system may be extended as accurate information about the current state of the software is made available.

The domain of problems our research addressed included looking into the various weaknesses and vulnerabilities of software and associated documentation as the software ages.

Approach:

The approach of this effort was to explore and select methods of documentation that automatically extract and display program logic (and more) from source code, thereby extending the useful lifetime of software systems. The research concentrated on developing documentation directly from the source code, not from sources that had been modified for documentation generation purposes. (These modifications are often called "signals" embedded in the code.)

Goals:

The goals of the research were to:

1. Research techniques specifically appropriate to the software maintenance environment, as contrasted with methods commonly used in development. This research resulted in a comprehensive set of information requirements that were needed to create that documentation.

2. Determine the design of a general documentation language, focusing on being able to handle a selected class of languages, using the proposed approach (structured in the three categories of preprocessor, compiler, postprocessor).

3. Determine if the proposed approach could be implemented to automatically transform programming language source into documentation, without requiring special signaling in the source code.

These goals mapped cleanly to the multi-phase approach desired.

SUMMARY OF IMPORTANT RESULTS

Goals:

At this time, the end of the contract, we believe that we have met our goals as follows:

1. The research into documentation methodologies in use today successfully led to the knowledge of the documentation requirements perceived relevant to and by maintenance programmers.

2. A general documentation grammar was developed, containing all features of the selected class of target languages. This grammar was implemented, i.e. a compiler was built for it, as the centerpiece to a prototype maintenance tool.

3. The 3-phase design was implemented; the DL compiler is essentially complete, the postprocessor for modified NS Diagrams is complete, and research and design for a FORTRAN to DL preprocessor is essentially complete. As such, the prototype accepts working C code and generates NS Diagrams for it.

Important Results:

Work on this project led to some interesting conclusions. The first of these was the determination that none of the methodologies in use today is really optimal in its utility to maintenance programmers. While NS Diagrams and others are familiar to programmers and to many knowledgeable users, the optimal tool would provide a new methodology that provided graphics, text, detail, overview, static structure, and dynamic dataflow all in a visually-appealing, user-friendly way.

The second conclusion is that the generality requirement is indeed a requirement, not just a consideration of interest. Programmers working with different languages have widely different perceptions about what is useful documentation. Thus a documentation tool that provided only one type of documentation would not solve the problem.

PUBLICATIONS AND TECHNICAL REPORTS PUBLISHED

A paper titled "Documentation in a Software Maintenance Environment" was published by, and presented at, the Conference on Software Maintenance (CSM-88).

The project Interim Technical Report, "Documentation in a Software Maintenane Environment", is available through DTIC, AD number: ADA 185 892.

The project Final Technical Report, "Automatic Documentation Methodologies for Software Maintenance", is nearing completion and will be submitted for inclusion in DTIC holdings.

PARTICIPATING SCIENTIFIC PERSONNEL

The following technical personnel were supported in part or in full by this effort:

Andrew J. Fine
Alton L. Gilbert
Michael Hamilton
William L. Hembree
Patricia M. Hyland
Larry D. Landis
Mona McWilliams
Robert Melson
W. Wes Weeks
Keith Whalen

This concludes the project Final Report.